UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/816,103	03/31/2004	Edward T. Grochowski	42P18305	9899

8791 7590 03/05/2007

BLAKELY SOKOLOFF TAYLOR & ZAFMAN
12400 WILSHIRE BOULEVARD
SEVENTH FLOOR
LOS ANGELES, CA 90025-1030

EXAMINER	
FENNEMA, ROBERT E	
ART UNIT	PAPER NUMBER
2183	

SHORTENED STATUTORY PERIOD OF RESPONSE	MAIL DATE	DELIVERY MODE
3 MONTHS	03/05/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

If NO period for reply is specified above, the maximum statutory period will apply and will expire 6 MONTHS from the mailing date of this communication.

Office Action Summary	Application No.	Applicant(s)	
	10/816,103	GROCHOWSKI ET AL.	
	Examiner	Art Unit	
	Robert E. Fennema	2183	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *12 December 2006*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-18,20-28,32-51 and 55-68* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-18,20-28,32-51 and 55-68* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date *9/12/2006*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-18, 20-28, 32-51, 55-68 have been considered. Claims 66-68 added as per Applicant's request. Claims 1, 3, 17, 23, 35, 44, 55, and 61 amended as per Applicant's request.

Claim Rejections - 35 USC § 102

2. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

3. Claims 17-18, 20-28 and 32 are rejected under 35 U.S.C. 102(b) as being anticipated by Lawlor et al. (USPN 5,485,626, herein Lawlor).

4. As per Claim 17, Lawlor teaches: An apparatus, comprising:

execution resources to execute a plurality of instructions (Column 10, Lines 47 and 51, the processors),

the execution resources to receive a non-privileged user instruction (Column 8, Lines 37-42. Furthermore, the disclosed invention is geared towards not using the operating system (Abstract and Column 6, Lines 38-42), meaning that these instructions would be non-privileged);

the execution resources further to, responsive to the received instruction, begin execution of a shred concurrently with one or more other shreds (Column 12, Lines 31-32); and

a shared register that is addressable by a user-level instruction, the shared register to provide communication between the shred and the one or more other shreds (Column 19, Lines 31-32).

5. As per Claim 18, Lawlor teaches: The apparatus of claim 17, further comprising:

one or more shared shred registers to facilitate communication between two or more of the shreds (Column 19, Lines 31-32).

6. As per Claim 20, Lawlor teaches the apparatus of claim 18, wherein the one or more shared registers further comprise a first register that enables an operating system or BIOS to enable multithreading architecture extensions for user-level multithreading (Column 18, Lines 13-19).

7. As per Claim 21, Lawlor teaches: The apparatus of claim 17, wherein the execution resources are further to, responsive to the received instruction, begin execution of a shred concurrently with one or more other shreds, without control of an operating system (Column 12, Lines 31-32).

8. As per Claim 22, Lawlor teaches: The apparatus of claim 17, wherein the execution resources include one or more processor cores capable of executing multiple shreds concurrently (Column 10, Lines 47 and 51).

9. As per Claim 23, Lawlor teaches: The apparatus of claim 17, further comprising:

One or more registers to hold a state shared among the shred and the one or more other shreds (Column 8, Lines 20-24, they share an object space containing counters and queues, which can be implemented in registers as seen in Column 8, Lines 59-60).

10. As per Claim 24, Lawlor teaches: The apparatus of claim 17, wherein the shred and the one or more other shreds share a current privilege level and share a common address translation (Column 8, Lines 37-42 show that the compiler creates instructions to create parallelism (threads/shreds) outside of the operating system, giving them the same privilege level (non privileged) as the others. In addition, Column 8, Lines 20-23 discusses the addressing).

11. As per Claim 25, Lawlor teaches: The apparatus of claim 17, further comprising logic to execute a user-level instruction to create the shred (Column 8, Lines 37-42).

12. As per Claim 26, Lawlor teaches: The apparatus of claim 17, further comprising a mechanism to perform communication between the shred and the one or more other

(Column 17, Lines 13-18).

13. As per Claim 27, Lawlor teaches: The apparatus of claim 17, further comprising sharing a system state among the shred and the one or more other shreds (Column 8, Lines 20-28).

14. As per Claim 28, Lawlor teaches: The apparatus of claim 26, wherein the mechanism further comprises one or more shared registers (Column 19, Lines 31-32).

15. As per Claim 32, Lawlor teaches: The apparatus of claim 17, further comprising: a user-level exception mechanism to report an exception to the shred (Column 18, Lines 21-22).

Claim Rejections - 35 USC § 103

16. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

17. Claims 1-12, 15-16, 35-44, 45-46, 50, and 62 are rejected under 35 U.S.C. 103(a) as being unpatentable over Lawlor, in view of Galvin et al. (herein Galvin)

18. As per Claim 1, Lawlor teaches: A method, comprising:

encountering a non-privileged user-level programming instruction (Column 8, Lines 37-42. Furthermore, the disclosed invention is geared towards not using the operating system (see Column 4, Lines 20-24 and Abstract), meaning that these instructions would be non-privileged, as seen in the Foldoc articles "privileged instruction" and "supervisor mode");

creating, responsive to the programming instruction, a first shared resource thread (shred) (Column 8, Lines 37-42, and Column 24, Line 36 as an example of a thread creation instruction); and

executing, responsive to the programming instruction, the first shred concurrently with at least one of the one or more other shreds (Column 12, Lines 31-32);

wherein creating the first shred is performed in hardware, without the intervention of an operating system (Abstract and Column 6, Lines 38-42), but fails to teach:

the shreds belonging to the same process as one or more other shreds.

While Lawlor teaches a plurality of threads/shreds, and that the threads can be executed concurrently with each other, he is silent towards the threads/shreds belonging to the same process as another thread/shred. However, Galvin teaches about processes and threads, in that a process/task contains one or more threads to execute the functionality of the process (see pages 111-115), and that this is the organization of how threads work (threads are part of a process/task, and that threads must be part of a process/task). Galvin also teaches on page 112 that the advantage of having multiple threads operate in a single process is that if one thread blocks, the other threads can run, leading to higher throughput and improved performance. Given these two

advantages, one of ordinary skill in the art would have been motivated to use the teachings of Galvin to utilize the threads disclosed in Lawlor to parallelize processes, where multiple threads reside in the same process.

19. As per Claim 2, Lawlor teaches: The method of claim 1, further comprising:

Maintaining a private state for the first thread, wherein the private state is associated with at least one of a plurality of registers including general purpose registers, floating point registers, MMX registers, segment registers, a flags register, an instruction pointer, control and status registers, SSE registers, and a MXCSR register (Column 18, Lines 14-18).

20. As per Claim 3, Lawlor teaches: The method of claim 1, further comprising:

sharing a state among a plurality of shared resource threads ("shreds") associated with a first operating-system-generated thread, the plurality of shreds including the first shred and the one or more other shreds; while not sharing said state with a second shred, created as a result of a second non-privileged user-level programming instruction that is associated with a second operating-system-generated thread (Column 11, Lines 3-5 show this context is stored in the SV register 112, where the operating system generated threads are processes as described by Galvin).

21. As per Claim 4, Lawlor teaches: The method of claim 1, further comprising:

sharing a state among a plurality of shreds of the one or more shreds; and storing the state in one or more registers (Column 8, Lines 20-24, they share an object space containing counters and queues, which can be implemented in registers as seen in Column 8, Lines 59-60).

22. As per Claim 5, Lawlor teaches: The method of claim 1, wherein the first shred and the one or more shreds share a current privilege level and share a common address translation (Column 8, Lines 37-42 show that the compiler creates instructions to create parallelism (threads/shreds) outside of the operating system, giving them the same privilege level (non privileged) as the others. In addition, Column 8, Lines 20-23 discusses the addressing).

23. As per Claim 6, Lawlor teaches: The method of claim 1, further comprising:

receiving a non-privileged user-level programming instruction that encodes a shred destroy operation (Column 29, Lines 27, 49, and Column 30, Line 4).

24. As per Claim 7, Lawlor teaches: The method of claim 1, further comprising communicating between the first shred and at least one of the one or more other shreds (Column 17, Lines 13-18).

25. As per Claim 8, Lawlor teaches: The method of claim 1, further comprising sharing a system state among the one or more shreds (Column 8, Lines 20-28).

26. As per Claim 9, Lawlor teaches: The method of claim 7, wherein said communicating is performed via one or more shared registers (Column 19, Lines 31-32).

27. As per Claim 10, Lawlor teaches: The method of claim 1, further comprising:

scheduling, responsive to a user-level programming instruction, the first shred and the one or more other shreds without intervention of the operating system (Column 7, Lines 37-42).

28. As per Claim 11, Lawlor teaches: The method of claim 7, wherein:

said communicating is performed via a user-level shred signaling instruction (Column 17, Lines 13-18).

29. As per Claim 12, Lawlor teaches: The method of claim 11, further comprising:

storing one or more shred states associated with the one or more shreds responsive to receipt of a context switch request (Column 18, Lines 14-18).

30. As per Claim 15, Lawlor teaches: The method of claim 1, wherein:

The shred is to perform input/output (I/O) operations (Column 12, Lines 35-41).

31. As per Claim 16, Lawlor teaches: The method of claim 1, wherein:

the one or more shreds are to perform computation functions (Column 12, Lines 31-32).

32. As per Claim 35, Lawlor teaches: An article of manufacture, comprising:

a machine-accessible medium including data that, when accessed by a machine, cause the machine to perform operations comprising,

executing the plurality of threads of execution concurrently on multiple instruction sequencers (Column 12, Lines 31-32), but fails to teach:

receiving user-level programming instructions to execute a plurality of threads of execution that each shares a system state with an OS-generated thread.

Lawlor teaches user-level instructions to execute a plurality of threads, however, Lawlor is silent towards the threads sharing a system state with an OS-generated thread, although Lawlor does teach of operating systems which support threading. However, Galvin teaches about processes and threads, and that a process (or task) is created by the operating system, and that threads are spawned for each process, with their own private states, but also share a common state (the address space of the process, see Pages 111-115), and these processes are created by the operating system (Page 106). Given that a process is essentially a thread, one could consider a process/task to be an operating-system generated thread, and given the need to know

how to implement an OS with threading, without Lawlor providing the specifics and high-level implementation, one of ordinary skill in the art would have been motivated to combine the teachings of Galvin into Lawlor's invention in order to see how to use an OS with the invention properly, and would have incorporated the knowledge of processes into the invention (which are also extremely well known in the art of multithreading), thus the user-level threads created by Lawlor would share a state with the process upon which they are created from.

33. As per Claim 36, Lawlor teaches: The article of manufacture of claim 35, wherein the operations further comprise:

maintaining a private state for each shred of the plurality of shreds, wherein the private state is associated with at least one of a plurality of registers including general purpose registers, floating point registers, MMX registers, segment registers, a flags register, an instruction pointer, control and status registers, SSE registers, and a MXCSR register (Column 18, Lines 14-18).

34. As per Claim 37, Lawlor teaches: The article of manufacture of claim 35, wherein the machine-accessible medium further includes data that causes the machine to perform operations comprising sharing a first state among the plurality of shreds, while maintaining a second state privately among an additional shred associated with a thread that is not associated with the plurality of shreds, wherein the first state is associated with at least one of a plurality of registers including a control register, a flags

register, memory management registers, a local descriptor table register, a task register, debug registers, model specific registers, shared registers, and shred control registers (Column 11, Lines 3-5 show the program context is stored in the SV register 112).

35. As per Claim 38, Lawlor teaches: The article of manufacture of claim 35, wherein the machine-accessible medium further includes data that causes the machine to perform operations comprising:

sharing a state among the plurality of shreds; and storing the state in one or more registers (Column 8, Lines 20-24, they share an object space containing counters and queues, which can be implemented in registers as seen in Column 8, Lines 59-60).

36. As per Claim 39, Lawlor teaches: The article of manufacture of claim 35, wherein the plurality of shreds share a current privilege level and share a common address translation (Column 8, Lines 37-42 show that the compiler creates instructions to create parallelism (threads/shreds) outside of the operating system, giving them the same privilege level (non privileged) as the others. In addition, Column 8, Lines 20-23 discusses the addressing).

37. As per Claim 40, Lawlor teaches: The article of manufacture of claim 35, wherein the one or more user-level programming instructions include an instruction to create one

or more of the plurality of shreds (Column 8, Lines 37-42).

38. As per Claim 41, Lawlor teaches: The article of manufacture of claim 35, wherein the machine-accessible medium further includes data that causes the machine to perform operations comprising communicating among the plurality of shreds (Column 17, Lines 13-18).

39. As per Claim 42, Lawlor teaches: The article of manufacture of claim 35, wherein the machine-accessible medium further includes data that causes the machine to perform operations comprising sharing a system state among the plurality of shreds (Column 8, Lines 20-28).

40. As per Claim 43, Lawlor teaches: The article of manufacture of claim 35, wherein the machine-accessible medium further includes data that causes the machine to perform operations comprising communicating between the plurality of shreds via one or more shared registers (Column 19, Lines 31-32).

41. As per Claim 44, Lawlor teaches: The article of manufacture of claim 35, wherein an application program controls the plurality of shreds directly, including scheduling of the plurality of shreds, and wherein an operating system executed by the multiprocessor schedules one or more threads (Column 2, Lines 21-31).

42. As per Claim 50, Lawlor teaches: The article of manufacture of claim 35, wherein the plurality of shreds perform input/output (I/O) functions and computation functions (Column 12, Lines 31-41).

43. As per Claim 62, Lawlor teaches: The article of manufacture of claim 35, wherein the one or more user-level programming instructions include an instruction to destroy one or more of the plurality of shreds (Column 29, Lines 27, 49, and Column 30, Line 4).

44. Claims 45-46 are rejected under 35 U.S.C. 103(a) as being unpatentable over Lawlor and Galvin, further in view of Foldoc.

45. As per Claim 45, Lawlor teaches: The article of manufacture of claim 35, wherein the machine-accessible medium further includes data that causes the machine to perform operations comprising:

associating the plurality of shreds with a thread (Column 8, Lines 20-24), but fails to teach:

suspending the plurality of shreds belonging to the thread when a context switch request is received through a single one of the plurality of shreds. In Column 18, Lines 13-18, Lawlor teaches that the context is saved when a thread goes inactive, but does not teach that all threads/shreds of a process would be made to be suspended on a context switch. However, Foldoc (Context switch) teaches that on a context switch between processes, the entire process stops running, and another begins. Given that

the reason to multithread is to maximize the shared state as much as possible, it would have been obvious to one of ordinary skill in the art at the time the invention was made to shut down all the shreds associated with the thread/process when it was switched out, so the new process could make use of the advantages of parallelism without the other shreds getting in the way.

46. As per Claim 46, Lawlor teaches: The article of manufacture of claim 45, wherein the machine-accessible medium further includes data that causes the machine to perform operations comprising:

storing one or more shred states associated with the plurality of shreds when the context switch request is received (Column 18, Lines 13-18).

47. Claims 13-14 are rejected under 35 U.S.C. 103(a) as being unpatentable over Lawlor and Foldoc, further in view of Golliver et al. (USPN 6,378,067, herein Golliver).

48. As per Claim 13, Lawlor teaches the method of claim 1, but fails to teach:

handling with user-level exception handler code an exception generated during execution of the first shred, without intervention of the operating system.

Lawlor teaches that exceptions can be generated and are dealt with (Column 18, Lines 21-22 for example), but does not explicitly disclose how this occurs, or what deals with the exceptions. Golliver teaches a method to handle multiple exceptions at once by prioritizing them, and dealing with them in a non-operating system exception handler

(Column 3, Line 66 – Column 4, Line 6, and Column 4, Lines 22-25. The operating system is not used to process the exception unless explicitly called, as shown in Column 7, Lines 22-27). Given the need for a method to handle exceptions, and the desire to minimize the operating systems involvement in thread processing in Lawlor's invention, and additionally the need to handle multiple exceptions which could come up when executing multiple shreds concurrently, it would have been obvious to one of ordinary skill in the art at the time the invention was made to use Golliver's exception handler to handle Lawlor's exceptions.

49. As per Claim 14, Golliver teaches: The method of claim 13, further comprising:

receiving the exception from an application program (Column 4, Lines 22-26); and

determining whether to report the exception to the operating system (Column 7, Lines 25-27).

50. Claims 33-34 and 59-61 are rejected under 35 U.S.C. 103(a) as being unpatentable over Lawlor, in view of Golliver.

51. As per Claim 33, Lawlor teaches the apparatus of claim 17, but fails to teach:

an exception mechanism to report an exception to an operating system.

Lawlor teaches that exceptions can be generated and are dealt with (Column 18, Lines 21-22 for example), but does not explicitly disclose how this occurs, or what deals

with the exceptions. Golliver teaches a method to handle multiple exceptions at once by prioritizing them, and dealing with them in a non-operating system exception handler (Column 3, Line 66 – Column 4, Line 6, and Column 4, Lines 22-25. The operating system is not used to process the exception unless explicitly called, as shown in Column 7, Lines 22-27). Given the need for a method to handle exceptions, and the desire to minimize the operating systems involvement in thread processing in Lawlor's invention (but still be able to use it when necessary), and additionally the need to handle multiple exceptions which could come up when executing multiple shreds concurrently, it would have been obvious to one of ordinary skill in the art at the time the invention was made to use Golliver's exception handler to handle Lawlor's exceptions.

52. As per Claim 34, Golliver teaches: The apparatus of claim 32, further comprising:

a mechanism to detect multiple exceptions, each exception associated with a different one of a plurality of concurrently-executing shreds, where the plurality includes the shred and the one or more other shreds (Column 3, Line 66 – Column 4, Line 6, and Column 5, Lines 29-31);

wherein the exception mechanism includes a prioritizer to prioritize the exceptions (Column 4, Line 1); and

wherein the exception mechanism is further to report only one of the prioritized exceptions at a time to the operating system (Column 7, Lines 23-28 disclose the operating system may be able to be told to handle exceptions, and Column 7, Lines 11-13 show that the faults can be reported one at a time).

53. As per Claim 59, Golliver teaches: The apparatus of claim 32, wherein: the user-level exception mechanism is further to vector to a fixed location in order to allow the shred to service to the exception (Figure 4, and Column 7, Lines 23-28, the user handler set).

54. As per Claim 60, Golliver teaches: The apparatus of claim 32, wherein:
the plurality of instructions further include a system call instruction to explicitly invoke an operating system to service to the exception (Column 7, Lines 23-28).

55. As per Claim 61, Golliver teaches: The apparatus of claim 34, wherein said prioritizer employs a round-robin scheme (Column 7, Lines 11-12 and Column 5, Lines 30-32).

56. Claims 47-49 are rejected under 35 U.S.C. 103(a) as being unpatentable over Lawlor and Galvin, further in view of Golliver.

57. As per Claim 47, Lawlor teaches the article of manufacture of claim 35, but fails to teach:

wherein the machine-accessible medium further includes data that causes the machine to perform operations comprising:

reporting one or more exceptions to a first shred of the plurality of shreds.

Lawlor teaches exceptions occurring (Column 18 Lines 20-21 for example), but does not teach explicitly what happens during an exception. Golliver teaches a method to handle multiple exceptions at once by prioritizing them, and reporting them to a non-operating system exception handler (Column 3, Line 66 – Column 4, Line 6, and Column 4, Lines 22-25. The operating system is not used to process the exception unless explicitly called, as shown in Column 7, Lines 22-27). Given the need for a method to handle exceptions, and the desire to minimize the operating systems involvement in thread processing in Lawlor's invention (but still be able to use it when necessary), and additionally the need to handle multiple exceptions which could come up when executing multiple shreds concurrently, it would have been obvious to one of ordinary skill in the art at the time the invention was made to use Golliver's exception handler to handle Lawlor's exceptions.

58. As per Claim 48, Golliver teaches: The article of manufacture of claim 47, wherein the machine-accessible medium further includes data that causes the machine to perform operations comprising:

reporting the one or more exceptions from an application program (Column 2, Lines 42-47); and

determining whether to report the one or more exceptions to an operating system (Column 7, Lines 23-27).

59. As per Claim 49, Golliver teaches: The article of manufacture of claim 48, wherein the machine-accessible medium further includes data that causes the machine to perform operations comprising:

prioritized-reporting of the one or more exceptions to the operating system; comprising receiving the one or more exceptions concurrently via different shreds of the plurality of shreds (Column 5, Lines 30-32); and

servicing one of the one or more exceptions according to the prioritized-reporting while suspending exception processing of other exceptions of the one or more exceptions (Column 7, Lines 11-13).

60. Claims 51, 55-58, and 63-66 are rejected under 35 U.S.C. 103(a) as being unpatentable over Lawlor, in view of Patterson et al. (herein Patterson).

61. As per Claim 51, Lawlor teaches: A system, comprising:

a microprocessor implementing an instruction set architecture (ISA) (Column 7, Lines 1-4); and

a memory (Figure 2, Memory 211 and cache 201); wherein the ISA includes one or more instructions to allow user-level multithreading operations (Column 7, Lines 1-4), but fails to teach:

the microprocessor capable of executing multiple concurrent shreds.

Lawlor teaches a multiprocessor system, where each processor is capable of working on a single shred at a time, but does not teach that each individual processor is

capable of executing more than one shred at a time. However, Patterson teaches a system of increasing parallelism and increasing performance by making a processor superscalar (Page 215). Instructions that do not depend on each other (i.e., independent instructions that can be run in parallel, just like shreds) can be issued simultaneously (Pages 216-217). Given this advantage, and the statement by Lawlor which says that being able to dynamically assign processors to increase the processing time of the job is favorable (Column 8, Lines 50-58), it would have been obvious to one of ordinary skill in the art at the time the invention was made to consider making Lawlor's processors superscalar, to further increase the parallelism of the shred processing.

62. As per Claim 55, Lawlor teaches: A system, comprising:

a microprocessor (Column 10, Line 58, Processor 106), including a plurality of user-level multithreading registers wherein the registers are addressable by one or more user-level instructions in each of a plurality of user-level threads and are to support communication among the user-level threads (Column 19, Lines 31-32); and

memory coupled to the microprocessor, the memory to store the one or more user-level instructions (Column 11, Lines 44-46), but fails to teach:

wherein the microprocessor is further to execute the user-level threads concurrently.

Lawlor teaches a multiprocessor system where each processor can work on a single thread/shred at a time, but does not teach that each individual processor is

capable of executing more than one thread/shred at a time. However, Patterson teaches a system of increasing parallelism and increasing performance by making a processor superscalar (Page 215). Instructions that do not depend on each other (i.e., independent instructions that can be run in parallel, just like shreds) can be issued simultaneously (Pages 216-217). Patterson further teaches that this superscalar machine can be used to execute multiple threads simultaneously (Pages 608-611), since modern processors have more functional unit parallelism than a single thread can exploit. Given these advantages, and the statement by Lawlor which says that being able to dynamically assign processors to increase the processing time of the job is favorable (Column 8, Lines 50-58), it would have been obvious to one of ordinary skill in the art at the time the invention was made to consider making Lawlor's processors superscalar, to further increase the parallelism of the shred processing.

63. As per Claim 56, Lawlor teaches: The system of claim 55, wherein the plurality of user-level multithreading registers further comprises a plurality of shared shred registers to facilitate communication between a plurality of shreds and to facilitate synchronization between the plurality of shreds (Column 19, Lines 31-32).

64. As per Claim 57, Lawlor teaches: The system of claim 56, wherein the plurality of user-level multithreading registers further comprises a plurality of shred control registers to manage the plurality of shreds (Column 18, Lines 13-18).

65. As per Claim 58, Lawlor teaches The system of claim 57, wherein the microprocessor:

receives programming instructions to execute one or more shreds in accordance with the ISA (Column 8, Lines 37-42);

configures one or more instruction sequencers via the ISA (Column 11, Lines 40-42); and

executes the one or more shreds concurrently (Column 12, Lines 31-32).

66. As per Claim 63, Lawlor teaches: The system of claim 51, wherein the one or more instructions include an instruction to create a shred without intervention of an operating system (Column 4, Lines 20-23).

67. As per Claim 64, Lawlor teaches: The system of claim 51, wherein the one or more instructions include an instruction to destroy a shred without intervention of an operating system (Column 29, Lines 27, 49, and Column 30, Line 4).

68. As per Claim 65, Lawlor teaches: The system of claim 51, wherein: the user-level multi-threading operations include concurrent execution of two or more shreds associated with the same thread (Column 4, Lines 10-15, where it says that a shred (fork) is created when a part of the program is found to be able to be executed in parallel, thus all the shreds are threads of an overall process (the thread in this case)).

69. As per Claim 66, Lawlor teaches: The system of Claim 55, wherein the memory is from a plurality of memory devices including DRAM, flash, and EEPROM (Column 11, Lines 44-46).

70. Claims 67-68 are rejected under 35 U.S.C. 103(a) as being unpatentable over Lawlor and Patterson, further in view of Galvin.

71. As per Claim 67, Lawlor teaches: A processor, comprising:

execution resources to execute instructions of an instruction set architecture (ISA), (Column 10, Lines 47 and 51),

wherein the ISA includes one or more non-privileged instructions to allow multithreading operations (Column 8, Lines 37-42), but fails to teach:

the execution resources to execute multiple concurrent shared resource threads ("shreds") that share application state.

Lawlor teaches of processors containing execution resources, with instructions to allow multithreading operations, but teaches that each processor in a multi-processor system can execute one thread or instruction at a time, and does not teach that a single processor can execute multiple threads concurrently. However, Patterson teaches a system of increasing parallelism and increasing performance by making a processor superscalar (Page 215). Instructions that do not depend on each other (i.e., independent instructions that can be run in parallel, just like shreds) can be issued simultaneously (Pages 216-217). Patterson further teaches that this superscalar

machine can be used to execute multiple threads simultaneously (Pages 608-611), since modern processors have more functional unit parallelism than a single thread can exploit. Given these advantages, and the statement by Lawlor which says that being able to dynamically assign processors to increase the processing time of the job is favorable (Column 8, Lines 50-58), it would have been obvious to one of ordinary skill in the art at the time the invention was made to consider making Lawlor's processors superscalar, to further increase the parallelism of the shred processing.

However, these combinations do not teach that the threads being executed concurrently share application state. Galvin teaches about processes and threads, in that a process/task contains one or more threads to execute the functionality of the process (see pages 111-115), and that this is the organization of how threads work (threads are part of a process/task, and that threads must be part of a process/task). Galvin also teaches on page 112 that the advantage of having multiple threads operate in a single process is that if one thread blocks, the other threads can run, leading to higher throughput and improved performance. Given these two advantages, one of ordinary skill in the art would have been motivated to use the teachings of Galvin to utilize the threads disclosed in Lawlor to parallelize processes, where multiple threads reside in the same process (where all threads in a process share address space).

72. As per Claim 68, Lawlor teaches: The processor of Claim 67, wherein:

the one or more non-privileged instructions include an instruction to create a shared resource thread (Column 8, Lines 37-42).

Response to Arguments

73. Applicant's arguments with respect to claims 17, 35, and 55 have been considered but are moot in view of the new ground(s) of rejection necessitated by Applicant's amendment.

However, in regards to Claims 17 and 55, it can be clearly seen in Column 19 of Lawlor that there are registers shared between threads used for communication between threads, and that these registers are addressable by user-level instructions, namely the SENDM and EQM instructions.

74. Applicant's arguments for Claim 1 are moot, as Applicant is arguing against a reference brought in to teach a limitation which Applicant has removed from the claim, which is why the reference explaining the definition of virtual memory does not teach the new limitation which has nothing to do with virtual memory.

75. Regarding Applicant's arguments with respect to Claim 51, Examiner has not found the argument persuasive. The measure for 103 is to determine if the combination of references would lead one of ordinary skill in the art to arrive at the claimed invention, and the combination of Lawlor and Patterson do just that. Lawlor teaches multiple processors each executing a single thread, but Lawlor states that the availability of processors to do the job can vary, and that as more become available, more should be

used to help, a clear indication that there is a required need for faster processing and increased parallelism, and while multiple processors can help, multiple processors are not always available to each execute a thread. Patterson teaches a superscalar processor, designed to increase parallelism by letting a processor execute more than one instruction at a time. With Lawlor teaching that multiple threads can be executed concurrently, with the limiting factor being each processor executing one instruction at a time, and Patterson teaching that a single processor can execute multiple instructions at a time, one of ordinary skill in the art would have been motivated to use these extra issue/execution slots to execute more threads, given the problems and solutions posed by Lawlor and Patterson. In addition, executing multiple threads concurrently on a single processor has been well known in the art for a very long time, and is not a patentable distinction without further detail.

In response to applicant's argument that the examiner's conclusion of obviousness is based upon improper hindsight reasoning, it must be recognized that any judgment on obviousness is in a sense necessarily a reconstruction based upon hindsight reasoning. But so long as it takes into account only knowledge which was within the level of ordinary skill at the time the claimed invention was made, and does not include knowledge gleaned only from the applicant's disclosure, such a reconstruction is proper. See *In re McLaughlin*, 443 F.2d 1392, 170 USPQ 209 (CCPA 1971).

Conclusion

76. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Robert E. Fennema whose telephone number is (571) 272-2748. The examiner can normally be reached on Monday-Friday, 8:00-5:30.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on (571) 272-4162. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Robert E Fennema
Examiner
Art Unit 2183

RF

EDDIE CHAN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100